Piensa Digital
<Tecnología real para soluciones reales/>

Capacitación formadores Generación de capacidades en el Ecosistema Digital de Bogotá

Formación especializada TI

Fedesoft MINTIC COLCIENCIAS GOBIERNO DE COLOMBIA

Piensa Digital
<Tecnología real para soluciones reales/>

CSS

Fedesoft MINTIC COLCIENCIAS GOBIERNO DE COLOMBIA

CSS - Cascading Style Sheets

Como HTML, CSS no es realmente un lenguaje de programación. Es un *lenguaje de hojas de estilo,* es decir, te permite aplicar estilos de manera selectiva a elementos en documentos HTML.

CSS Externo

Un CSS externo es cuando el CSS se encuentra en un archivo separado con una extensión .css

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>My CSS experiment</title>
    <link rel="stylesheet" href="style.css">
  </head>
  <body>
    <h1>Hello World!</h1>
    <p>This is my first CSS example</p>
  </body>
</html>
```

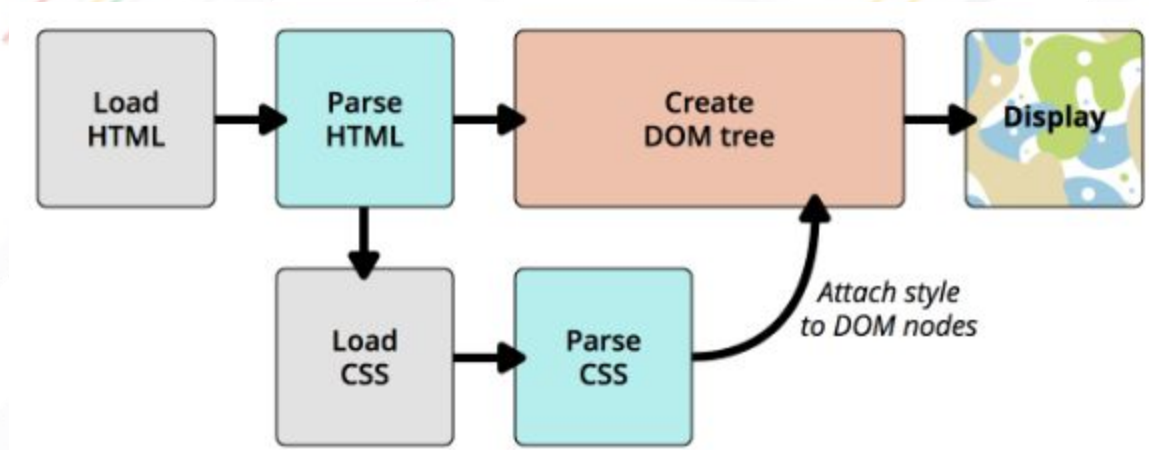
CSS Interno

En CSS interno no tenemos un CSS externo, sino que lo ubicamos dentro de un elemento <style>, en el apartado HTML head. Así, el HTML se escribirá:<html>

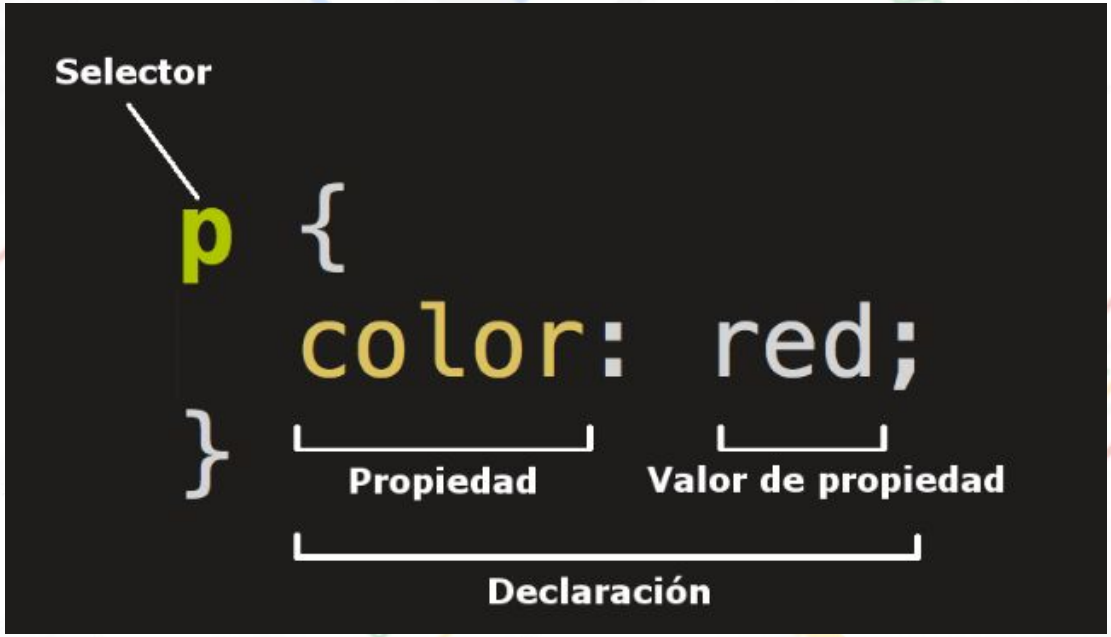
```
<head>
  <meta charset="utf-8">
  <title>My CSS experiment</title>
  <style>
    h1 {
      color: blue;
      background-color: yellow;
      border: 1px solid black;
    }
  </style>
</head>
```

DOM (Objeto Documento Modelo)

Este DOM representa el documento en la memoria del ordenador. Combinando el contenido del documento con su estilo.

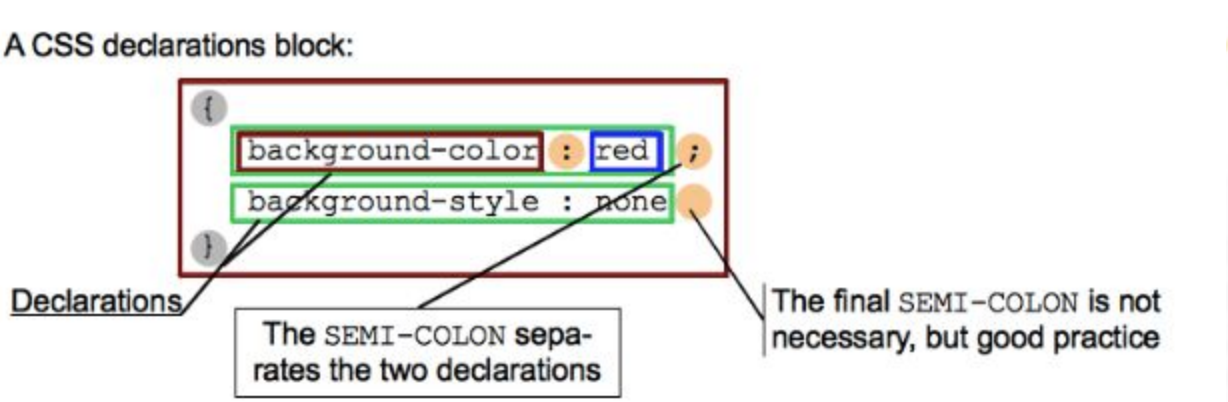


Anatomía de una regla CSS



Declaraciones en bloque

Las declaraciones están agrupadas por bloques, en los que el conjunto de declaraciones se encuentra enumerado entre llaves, uno inicial ({) y otro final (}).



Estilos en línea

Los estilos en una línea son declaraciones CSS que afectan solo a un elemento que está contenido dentro de un atributo style:

```
<body>
  <h1 style="color: blue;background-color: yellow;border: 1px solid black;">Hello World!</h1>
  <p style="color:red;">This is my first CSS example</p>
</body>
```

Selector

El elemento HTML en el que comienza la regla. esta selecciona el(los) elemento(s) a dar estilo (en este caso, los elementos p). Para dar estilo a un elemento diferente, solo cambia el selector.

Declaración

Una sola regla como color: red; especifica a cuál de las propiedades del elemento quieres dar estilo.

Propiedades

Maneras en las cuales puedes dar estilo a un elemento HTML. (En este caso, color es una propiedad del elemento p.) En CSS, seleccionas que propiedad quieres afectar en tu regla.

Valor de la propiedad

A la derecha de la propiedad, después de los dos puntos (:), tenemos el valor de la propiedad, para elegir una de las muchas posibles apariencias para una propiedad determinada (hay muchos valores para color además de red).

Ejercicio

Crear un folder llamado “clase 1”, crear un archivo index.html y style.css.

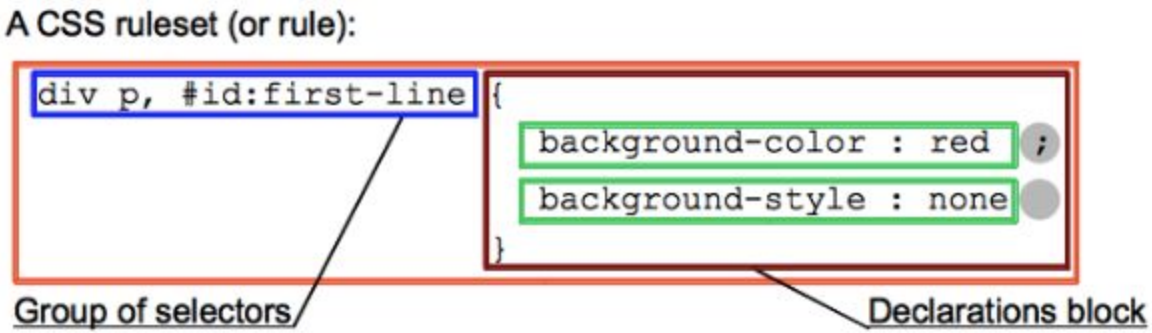
en el html deben generar la estructura básica y en el header llamar el css:

```
<link href="style.css" rel="stylesheet" type="text/css">
```

Agregue etiquetas la html con contenido y asignarles una regla en el archivo de CSS como demostración y que los chicos hagan lo mismo

SELECTORES

Los selectores forman parte de las reglas CSS y van justo antes de los bloques declarativos.



Tipos de Selectores

- Selectores simples
- Selectores de atributos
- Pseudo-elementos
- Combinaciones
- Selectores múltiples

Selectores simples

Este selector hace referencia directamente a un tipo de elemento HTML. Es la manera más sencilla para referirse a todos los elementos de un mismo tipo.

HTML

```
<p>What color do you like?</p>
<div>I like blue.</div>
<p>I prefer red!</p>
```

CSS

```
/* Todos los elementos p son rojos */
p {
  color: red;
}

/* Todos los elementos div son azules
*/
div {
  color: blue;
}
```

SELECTORES DE CLASES

El selector de clase se forma con un punto, '.', seguido de un nombre de clase. Un nombre de clase puede ser cualquier valor sin espacios usado dentro de un atributo HTML class.

HTML

```
<ul>
     <li class="first done">Create an
HTML document</li>
     <li class="second done">Create a
CSS style sheet</li>
     <li class="third">Link them all
together</li>
</ul>
```

CSS

```
/* El elemento con la clase "first" está en
negrita */
.first {
  font-weight: bold;
}

/* Todos los elementos con la clase "done" están
tachados */
.done {
  text-decoration: line-through;
}
```

Selectores ID

El selector ID está formado por una almohadilla (#), seguida del nombre ID de determinado elemento.

HTML

```
<p id="polite"> — "Good morning."</p>
<p id="rude"> — "Go away!"</p>
```

CSS

```
#polite {
  font-family: cursive;
}

#rude {
  font-family: monospace;
  text-transform: uppercase;
}
```

Piensa Digital
<Tecnología real para soluciones reales/>

Crear una lista de supermercado en HTML Y usar selectores CSS

Fedesoft MINTIC COLCIENCIAS GOBIERNO DE COLOMBIA

Selectores de Atributos

- **[attr] :** Este selector 'seleccionará' todos los elementos que contengan el atributo attr, sin importar el valor que tenga.
- **[attr=val] :** Este, seleccionará los elementos con el atributo attr, pero solo aquello cuyo valor coincida con val.
- **[attr~=val]:** Este selector afectará a los elementos con el atributo attr, pero solo si el valor val está contenido en la lista de valores (separados por espacios) incluidos en el valor de attr, por ejemplo una de las clases contenida en una lista de clases (separadas por espacios).

Selectores de Atributos

HTML

```
Ingredients for my recipe: <i lang="fr-FR">Poulet basquaise</i>
<ul>
  <li data-quantity="1kg" data-vegetable>Tomatoes</li>
  <li data-quantity="3" data-vegetable>Onions</li>
  <li data-quantity="3" data-vegetable>Garlic</li>
  <li data-quantity="700g" data-vegetable="not spicy like chili">Red pepper</li>
  <li data-quantity="2kg" data-meat>Chicken</li>
  <li data-quantity="optional 150g" data-meat>Bacon bits</li>
  <li data-quantity="optional 10ml" data-vegetable="liquid">Olive oil</li>
  <li data-quantity="25cl" data-vegetable="liquid">White wine</li>
</ul>
```

Ingredients for my recipe: *Poulet basquaise*

- Tomatoes
- Onions
- Garlic
- Red pepper
- Chicken
- Bacon bits
- Olive oil
- White wine

CSS

```
/* All elements with the attribute "data-vegetable"
   are given green text */
[data-vegetable] {
  color: green;
}

/* All elements with the attribute "data-vegetable"
   with the exact value "liquid" are given a golden
   background color */
[data-vegetable="liquid"] {
  background-color: goldenrod;
}

/* All elements with the attribute "data-vegetable",
   containing the value "spicy", even among others,
   are given a red text color */
[data-vegetable~="spicy"] {
  color: red;
}
```

Pseudo-classes and pseudo-elements

Una pseudo-clase CSS es una palabra clave añadida al final de un selector, precedida por dos puntos (:), que se usa para especificar que desea aplicar estilo al elemento seleccionado, pero solo cuando se encuentra en un estado determinado.

Por ejemplo, es posible que desee darle un estilo a un elemento de enlace sólo cuando está sobrevalorado por el puntero del mouse, o una casilla de verificación cuando está deshabilitada o marcada, o un elemento que es el primer elemento secundario de su elemento primario en el árbol DOM.

```
:active
:any
:checked
:default
:dir()
:disabled
:empty
:enabled
:first
:first-child
:first-of-type
:fullscreen
:focus
:focus-within
:hover
:indeterminate
:in-range
:invalid
:lang()
:last-child
:last-of-type
:left
:link
:not()
:nth-child()
:nth-last-child()
:nth-last-of-type()
:nth-of-type()
:only-child
:only-of-type
:optional
:out-of-range
:read-only
:read-write
:required
:right
:root
:scope
:target
:valid
:visited
```

Pseudo-classes and pseudo-elements

Piensa Digital
<Tecnología real para soluciones reales/>

CSS

```css
/* These styles will style our link
   in all states */
a {
  color: blue;
  font-weight: bold;
}

/* We want visited links to be the same color
   as non visited links */
a:visited {
  color: blue;
}

/* We highlight the link when it is
   hovered (mouse), activated
   or focused (keyboard) */
a:hover,
a:active,
a:focus {
  color: darkred;
  text-decoration: none;
}
```

HTML

```html
<a href="https://developer.mozilla.org/" target="_blank">Mozilla Developer Network</a>
```

Combinaciones

Combinación	Selecciona
A, B	Cualquier elemento seleccionado por A y/o B (ver Varios selectores en una regla, más adelante).
A B	Cualquier elemento seleccionado por B descendiente de un elemento seleccionado por A (o sea, un hijo, un hijo de otro hijo, etc.).
A > B	Cualquier elemento seleccionado por B y es hijo directo de un elemento seleccionado por A.
A + B	Cualquier elemento seleccionado por B y es el siguiente hermano de un elemento seleccionado por A (o sea, el siguiente hijo del mismo padre).
A ~ B	Cualquier elemento seleccionado por B y es uno de los siguientes hermanos del elemento seleccionado por A (uno de los siguientes hermanos del mismo padre).

Fuentes y texto

Recurso:

https://fonts.google.com/

Importar fuente:

```
<link href='http://fonts.googleapis.com/css?family=Open+Sans' rel='stylesheet' type='text/css'>
```

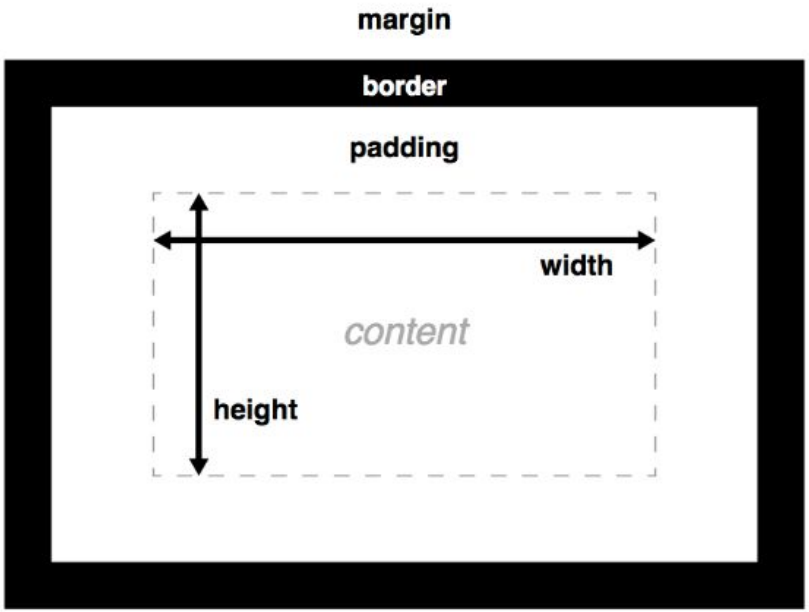
Usar fuente:

```
html {

font-family: 'Sunflower', sans-serif;
font-size: 10px; /* px quiere decir 'pixels': la base del tamaño de fuente es ahora de 10 pixeles*/
}
```

Cajas, cajas y ¡Más cajas!

Una cosa que notarás sobre la escritura de CSS es que trata mucho sobre cajas — ajustando su tamaño, color, posición, etc. Puedes pensar en la mayoría de los elementos HTML de tu página como cajas apiladas una sobre la otra.

Modelo de Cajas



Relleno(padding), el espacio alrededor del contenido (ej: alrededor del texto de un párrafo)

Marco(border), la línea que se encuentra fuera del relleno

Margen(margin), el espacio fuera del elemento que lo separa de los demás

Overflow

Cuando establece el tamaño de una caja con valores absolutos (por ejemplo, un **width/height** de píxel fijo), es posible que el contenido no se ajuste al tamaño permitido, en cuyo caso el contenido se desborda del cuadro. Para controlar lo que sucede en tales casos, podemos usar la propiedad de desbordamiento. Toma varios valores posibles, pero los más comunes son:

auto: si hay demasiado contenido, el contenido desbordado está oculto y las barras de desplazamiento se muestran para permitir al usuario desplazarse para ver todo el contenido.

oculto(hidden): si hay demasiado contenido, el contenido desbordado está oculto.

visible(visible): si hay demasiado contenido, el contenido desbordado se muestra fuera de la caja (este suele ser el comportamiento predeterminado).

Tipos de cajas (Display)

block: hace que el comportamiento del elemento sea como un bloque.

inline: el elemento se renderiza en línea con otros elementos.

inline-block: el elemento tendrá un comportamiento mezcla entre los dos anteriores, los elementos inline-block fluyen con el texto y demás elementos como si fueran elementos en-línea y además respetan el ancho, el alto y los márgenes verticales.

CSS layout

Las técnicas de diseño de página CSS nos permiten tomar los elementos contenidos en una página web y controlar dónde están posicionados en relación con su posición predeterminada en el flujo de diseño normal, los otros elementos a su alrededor, su contenedor principal o la viewport/window. Las técnicas de diseño de página que trataremos en más detalle en este módulo son:

- Floats
- Positioning
- CSS tables
- Flexbox
- Grid

Floats

Es una técnica que permite que los elementos floten hacia la izquierda o hacia la derecha el uno del otro, en lugar del valor predeterminado de posicionarse uno encima del otro. Los usos principales de los float son diseñar columnas y texto flotante alrededor de una imagen.

Valores: Left, Right, None, Inherit

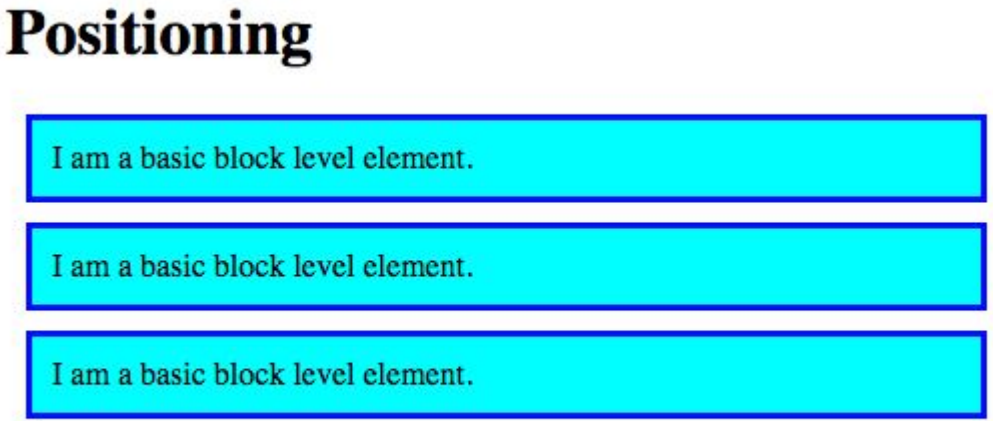
Técnicas de posicionamiento

El posicionamiento le permite mover un elemento desde su lugar original en la página a otro punto con gran precisión.

- El posicionamiento estático
- El posicionamiento relativo
- El posicionamiento absoluto
- El posicionamiento fijo

Posicionamiento simple

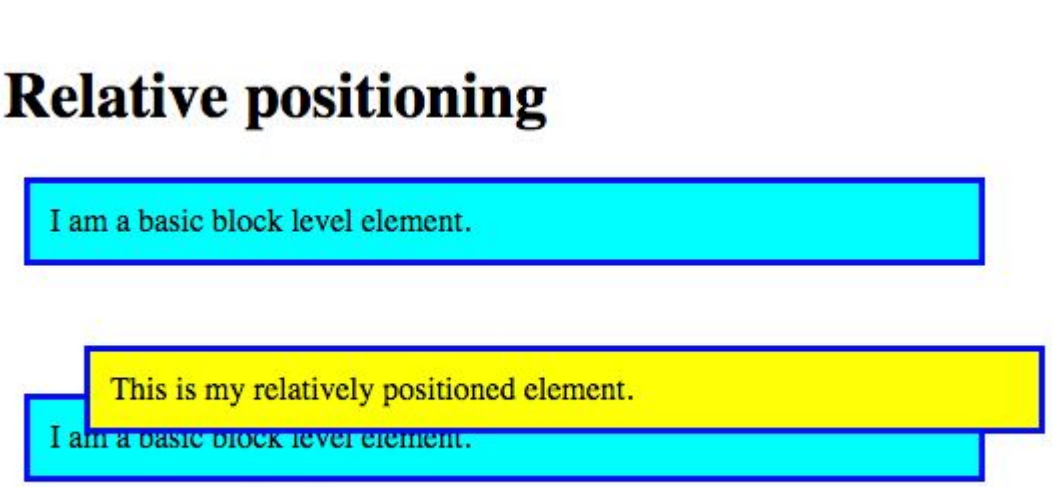
```
body {
  width: 500px;
  margin: 0 auto;
}

p {
  background: aqua;
  border: 3px solid blue;
  padding: 10px;
  margin: 10px;
}
```



Posicionamiento Relativo

El posicionamiento relativo desplaza una caja respecto de su posición original establecida mediante el posicionamiento normal. El desplazamiento de la caja se controla con las propiedades top, right, bottom y left.

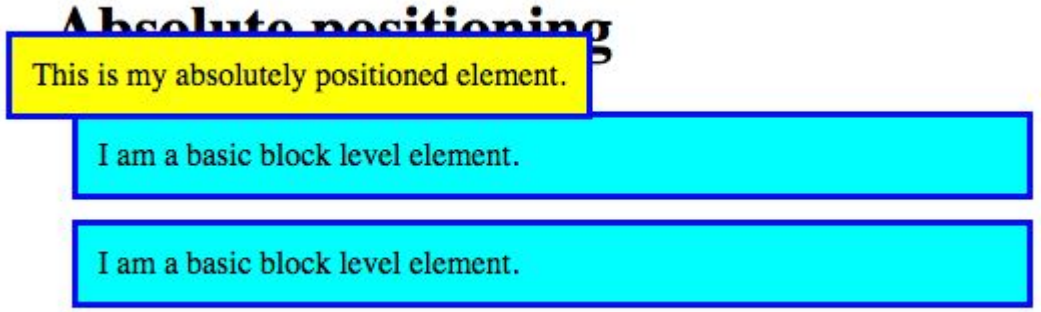
```
.positioned {
  position: relative;
  background: yellow;
  top: 30px;
  left: 30px;
}
```



Posicionamiento absoluto

El posicionamiento absoluto se emplea para establecer de forma exacta la posición en la que se muestra la caja de un elemento. La nueva posición de la caja se indica mediante las propiedades top, right, bottom y left.

```
.positioned {
  position: absolute;
  background: yellow;
  top: 30px;
  left: 30px;
}
```



Posicionamiento fijo

El estándar CSS considera que el posicionamiento fijo es un caso particular del posicionamiento absoluto, ya que sólo se diferencian en el comportamiento de las cajas posicionadas.

Cuando una caja se posiciona de forma fija, la forma de obtener el origen de coordenadas para interpretar su desplazamiento es idéntica al posicionamiento absoluto. De hecho, si el usuario no mueve la página HTML en la ventana del navegador, no existe ninguna diferencia entre estos dos modelos de posicionamiento.

Piensa Digital
<Tecnología real para soluciones reales/>

ESTRUCTURA DE TU WEB

Fedesoft MINTIC COLCIENCIAS GOBIERNO DE COLOMBIA

Estructura de archivos

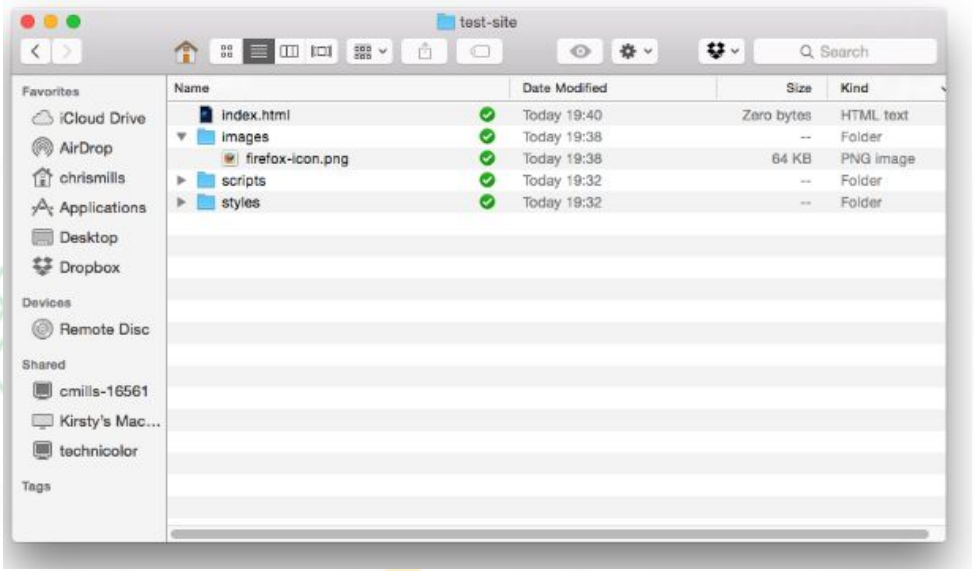
Veamos qué estructura debería tener nuestro sitio de prueba. Las cosas más comunes que tendremos en cualquier proyecto de sitio web que creamos son un archivo HTML índice y carpetas que contienen imágenes, archivos de estilo y archivos de script. Vamos a crear estos ahora:

index.html

Carpeta images

Carpeta styles

Carpeta scripts



Piensa Digital
<Tecnología real para soluciones reales/>

URL'S

Fedesoft MINTIC COLCIENCIAS GOBIERNO DE COLOMBIA

Para hacer que los archivos se comuniquen entre sí, tienes que proporcionarles una ruta entre ellos — básicamente es como un archivo se comunica con otro.

Algunas reglas generales para la ruta de Archivos:

Para enlazar un archivo que está en la misma carpeta que el archivo HTML, solamente usa el nombre del archivo, Ejemplo. `mi-imagen.jpg`.

Para hacer referencia a una subcarpeta, escribe en la ruta: primero el nombre de la carpeta y le agregas una diagonal, Ejemplo. `carpeta/mi-imagen.jpg`.

Para enlazar un archivo que está en el directorio **anterior** al que contiene al archivo HTML, deberás escribir dos puntos (..). Así por ejemplo, si `index.html` está dentro de una subcarpeta de `sitio-prueba` y `mi-imagen.png` está dentro de `sitio-prueba`, podrías referenciar tu archivo `mi-imagen.png` desde `index.html` usando `../mi-imagen.png`.

Puedes hacer muchas combinaciones así como, Por Ejemplo: `../subcarpeta/otra-subcarpeta/mi-imagen.png`.

Piensa Digital
<Tecnología real para soluciones reales/>

Clonemos:

https://www.modulosdesk.com/

Fedesoft MINTIC COLCIENCIAS GOBIERNO DE COLOMBIA